

FIG. 1

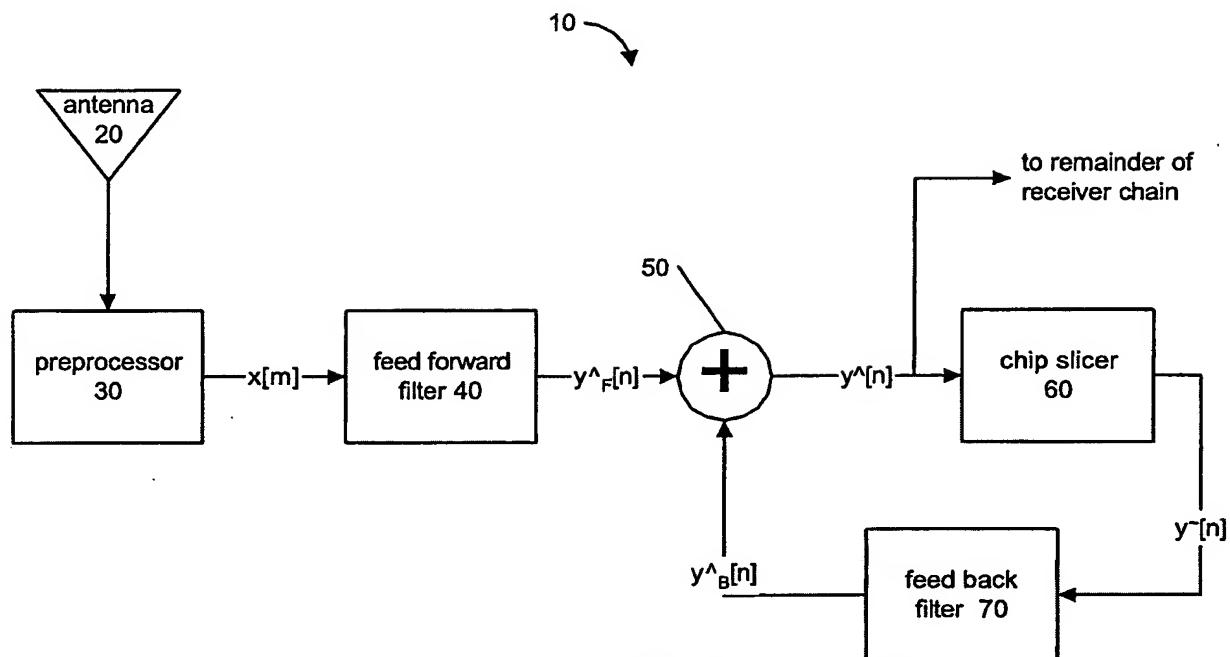


FIG. 2

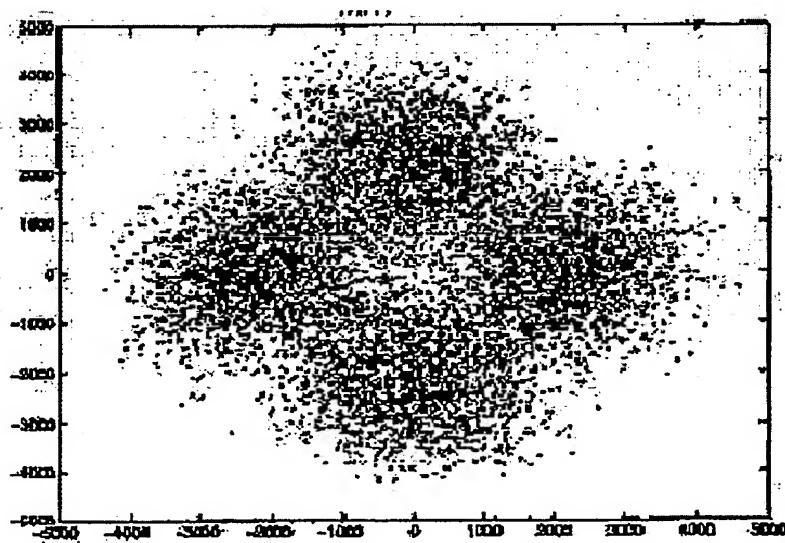


FIG. 3

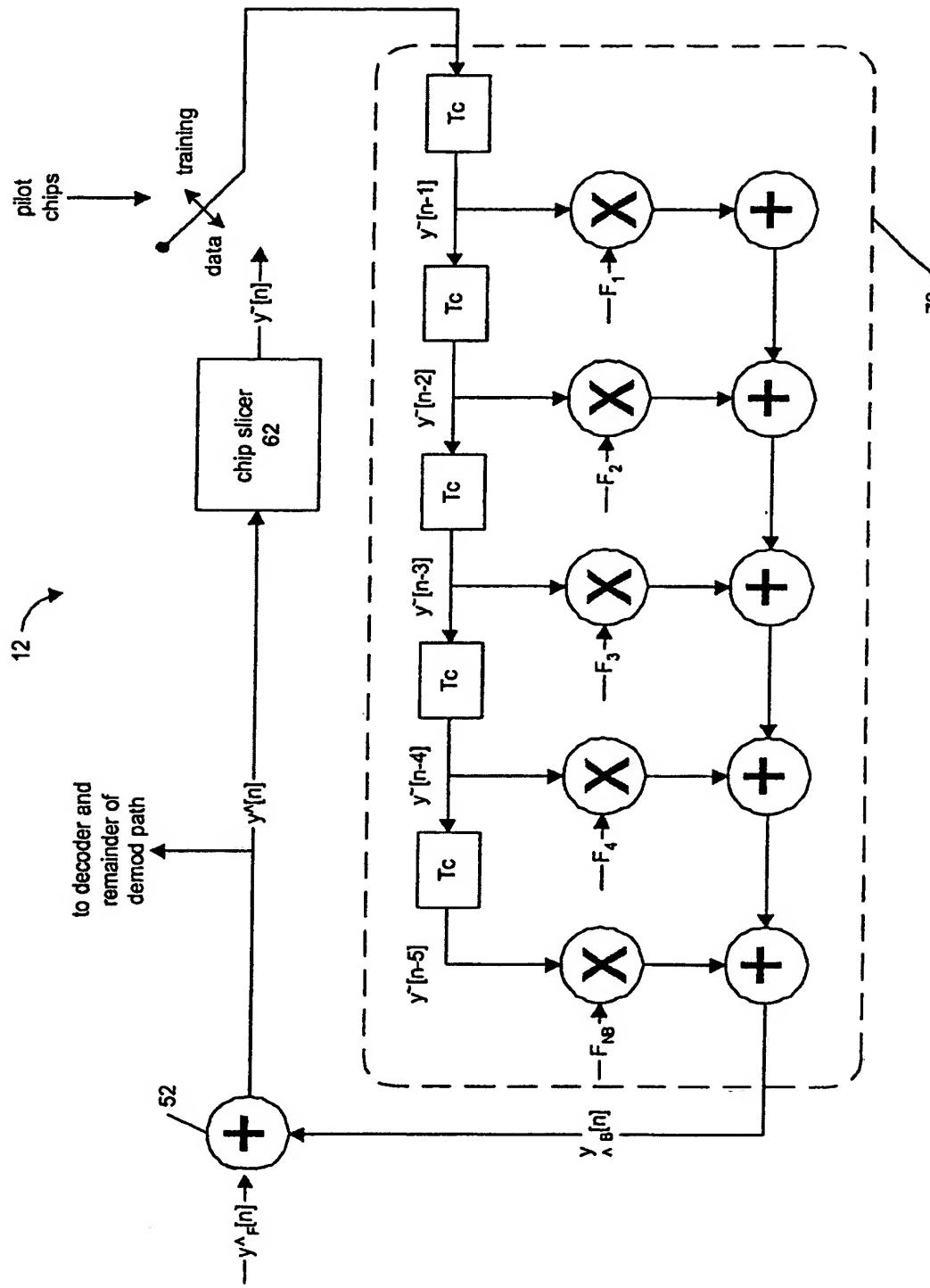


FIG. 4

content of FBF tap 9	content of FBF tap 8	content of FBF tap 7	content of FBF tap 6	content of FBF tap 5	content of FBF tap 4	content of FBF tap 3	content of FBF tap 2	content of FBF tap 1	slicer input at time (n)	CCK chip Index (k)	frame chip index (n)
									1	1	1
									2	2	2
							1	2	3with1	3	3
						1from3	2	3with1	4with2	4	4
					1from3	2from4	3with1	4with2	5	5	5
				1from3	2from4	3with1	4with2	5	6	6	6
			1from3	2from4	3with1	4with2	5	6	7with5	7	7
		1from3	2from4	3with1	4with2	5from7	6	7with5	8with6	8	8
	1from3	2from4	3with1	4with2	5from7	6from8	7with5	8with6	9	1	9
1from3	2from4	3with1	4with2	5from7	6from8	7with5	8with6	9	10	2	10
2from4	3with1	4with2	5from7	6from8	7with5	8with6	9	10	11with9	3	11
3with1	4with2	5from7	6from8	7with5	8with6	9from11	10	11with9	12with10	4	12
4with2	5from7	6from8	7with5	8with6	9from11	10from12	11with9	12with10	13	5	13
5from7	6from8	7with5	8with6	9from11	10from12	11with9	12with10	13	14	6	14
6from8	7with5	8with6	9from11	10from12	11with9	12with10	13	14	15with13	7	15
7with5	8with6	9from11	10from12	11with9	12with10	13from15	14	15with13	16with14	8	16
8with6	9from11	10from12	11with9	12with10	13from15	14from16	15with13	16with14	17	1	17
9from11	10from12	11with9	12with10	13from15	14from16	15with13	16with14	17	18	2	18
10from12	11with9	12with10	13from15	14from16	15with13	16with14	17	18	19with17	3	19
11with9	12with10	13from15	14from16	15with13	16with14	17from19	18	19with17	20with18	4	20
12with10	13from10	14from15	14from16	15with13	16with14	17from19	18from20	19with17	20with18	21	21

```
%%add new section to do 5.5M slicing based on CCK codeword knowledge (1=3, 2=-4, 5=-7, 6=8)
if MPDU_RATE==5.5 & kk==3
    slicer_output=hard_decision(real(chipCx1_estimate_vec(kk)+chipCx1_estimate_vec(kk-2)), ...
        imag(chipCx1_estimate_vec(kk)+chipCx1_estimate_vec(kk-2)));
elseif MPDU_RATE== 5.5 & kk==4
    slicer_output=hard_decision(real(chipCx1_estimate_vec(kk)-chipCx1_estimate_vec(kk-2)), ...
        imag(chipCx1_estimate_vec(kk)-chipCx1_estimate_vec(kk-2)));
elseif MPDU_RATE== 5.5 & kk==7
    slicer_output=hard_decision(real(chipCx1_estimate_vec(kk)-chipCx1_estimate_vec(kk-2)), ...
        imag(chipCx1_estimate_vec(kk)-chipCx1_estimate_vec(kk-2)));
elseif MPDU_RATE== 5.5 & kk==8
    slicer_output=hard_decision(real(chipCx1_estimate_vec(kk)+chipCx1_estimate_vec(kk-2)), ...
        imag(chipCx1_estimate_vec(kk)+chipCx1_estimate_vec(kk-2)));
end
```

FIG. 6A

```
If (kk==3 | kk==4 | kk==7 | kk==8) % use slicer output for improved DD-LMS on FFF
    FFF = FFF +LMS_stepsize*(FFF_content_I_Cx2+j*FFF_contentQ_Cx2)
    *conj(train_scale*slicer_output-chipCx1_estimate_vec(kk));

If (kk==3 | kk==4 | kk==7 | kk==8) % use slicer output for improved DD-LMS on FBF
    FBF = FBF +LMS_stepsize*(FBF_content1_Cx1+j*FBF_contentQ_Cx1)
    *conj(train_scale*slicer_output-chipCx1_estimate_vec(kk))
```

FIG. 6B

```
FBF_contentI_Cx1 = [FBF_input_scale*real(slicer_output); FBF_contentI_Cx1(1:end-1)];  
FBF_contentQ_Cx1 = [FBF_input_scale*imag(slicer_output); FBF_contentQ_Cx1(1:end-1)];  
  
%% Add Section to modify FBF contents based on more accurate sliced value. (i.e. on chips 3, 4, 7, 8  
%% we have better information on the FBF contents that are 2 chips old, so we replace FBF(3) with the  
%% appropriately signed version of FBF(1))  
  
if kk == 3  
    FBF_contentI_Cx1(3)=FBF_contentI_Cx1(1);  
    FBF_contentQ_Cx1(3)=FBF_contentQ_Cx1(1);  
elseif kk == 4  
    FBF_contentI_Cx1(3)=-FBF_contentI_Cx1(1);  
    FBF_contentQ_Cx1(3)=-FBF_contentQ_Cx1(1);  
elseif kk == 7  
    FBF_contentI_Cx1(3)=-FBF_contentI_Cx1(1);  
    FBF_contentQ_Cx1(3)=-FBF_contentQ_Cx1(1);  
elseif kk == 8  
    FBF_contentI_Cx1(3)=FBF_contentI_Cx1(1);  
    FBF_contentQ_Cx1(3)=FBF_contentQ_Cx1(1);  
end
```

FIG. 6C